

Remark

1. This Office action has been issued in response to amendment filed on 02/01/2011.

Claims 1-24 are pending.

EXAMINER'S AMENDMENT

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee. Authorization for this examiner's amendment was given in a telephone interview with Dalei Dong on 03/25/2011.

AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions, and listings, of claims in the application.

1. (Currently Amended) A computer-implemented system comprises at least a programmed computer processor for automated generation of one or more query language statements, the computer implemented system comprising:

a syntax pattern selector module using the programmed computer processor for selecting, in an automated process, a syntax pattern corresponding to a desired function defining a default syntax pattern provided to the syntax pattern selector module and a syntax standard for use in generating the one or more query language statements;

a statement assembly module using the programmed computer processor for populating the syntax pattern, in the automated process with an argument data set associated with parameters

that specify information of a desired data set and the desired function provided to the statement assembly module as part of the process of generating the one or more query language statements, wherein the arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements; and

whereby at least one query language statement having a tree query structure generated based at least in part on the parameters that specify information of the desired data set is assembled to be run against a data source to return the desired data set.

2. (Previously Presented) The computer-implemented system of claim 1, wherein the syntax pattern selector module selects the syntax pattern from a plurality of syntax patterns corresponding to a plurality of database management systems.

3. (Previously Presented) The computer-implemented system of claim 1, wherein the syntax pattern selector module selects the syntax pattern from a plurality of syntax patterns based upon at least one selection variable.

4. (Currently Amended) A computer-implemented system comprises at least a programmed computer processor for generating one or more query language statements, the computer implemented system comprising:

a syntax pattern selector module using the programmed computer processor for selecting a syntax pattern corresponding to a desired function defining a default syntax pattern, and a syntax standard;

a statement assembly module using the programmed computer processor for populating the syntax pattern with an argument data set associated with parameters that specify information of a desired data set and the desired function, wherein the arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements; a structure generator module using the programmed computer processor for generating a query structure based on the desired data set, the query structure providing a basis for identifying the desired function to be used by the syntax pattern selector module; and

whereby at least one query language statement having a tree query structure generated based at least in part on the parameters that specify information of the desired data set is assembled to be run against a data source to return the desired data set.

5. (Currently Amended) A computer-implemented system comprises at least a programmed computer processor for generating one or more query language statements, the computer implemented system comprising:

a syntax pattern selector module using the programmed computer processor for selecting a syntax pattern corresponding to a desired function defining a default syntax pattern, and a syntax standard;

a statement assembly module using the programmed computer processor for populating the syntax pattern with an argument data set associated with parameters that specify information of a desired data set and the desired function, wherein the arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements; a function

identifier module using the programmed computer processor for identifying a functional element corresponding to the desired function and at least one syntax pattern; and

whereby at least one query language statement having a tree query structure generated based at least in part on the parameters that specify information of the desired data set is assembled to be run against a data source to return the desired data set.

6. (Currently Amended) A computer-implemented system comprises at least a programmed computer processor for generating one or more query language statements, the computer implemented system comprising:

a syntax pattern selector module using the programmed computer processor for selecting a syntax pattern corresponding to a desired function defining a default syntax pattern, and a syntax standard;

a statement assembly module using the programmed computer processor for populating the syntax pattern with an argument data set associated with parameters that specify information of a desired data set and the desired function, wherein the arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements; an argument generator module using the programmed computer processor for identifying the argument data set associated with the desired data set; and

whereby at least one query language statement having a tree query structure generated based at least in part on the parameters that specify information of the desired data set is assembled to be run against a data source to return the desired data set.

7. (Previously Presented) The computer-implemented system of claim 6, wherein the argument generator module identifies the argument data set based upon a syntax description associated with the desired function.

8. (Previously Presented) The computer-implemented system of claim 1, wherein the system is a component in an online analytical processing system, a reporting system, a business intelligence system, or a data mining system.

9. (Previously Presented) The computer-implemented system of claim 1, further comprising a plurality of driver modules, each of the driver modules including at least one syntax pattern associated with a selected database management system.

10. (Currently Amended) A computer-implemented method of generating one or more query language statements to be run against a one or more data sources, comprising the steps, performed by a computer system, of:

selecting a syntax pattern corresponding to a desired function defining a default syntax pattern, provided as an input and a syntax standard for use in generating the one or more query language statements;

populating the syntax pattern with an argument data set associated with parameters that specify information of a desired data set and the desired function provided as an input identifying the data set on which to operate from the data source as part of generating the one or more query language statements, wherein the arguments data are arranged in a vector associated with a

functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements; and

wherein, the populated syntax pattern is used to generate one or more query language statements having a tree query structure generated based at least in part on the parameters that specify information of the desired data set runnable against one or more data sources to return the desired data result set.

11. (Previously Presented) The computer-implemented method of claim 10, wherein the step of selecting the syntax pattern includes selecting the syntax pattern from a plurality of syntax patterns corresponding to a plurality of database management systems.

12. (Previously Presented) The computer-implemented method of claim 10, wherein the step of selecting the syntax pattern includes selecting the syntax pattern from a plurality of syntax patterns based upon at least one selection variable.

13. (Previously Presented) The computer-implemented method of claim 10, further comprising the step of generating a query structure based on the desired data set, the query structure providing a basis for identifying the desired function to be used in selecting the syntax pattern.

14. (Currently Amended) A method of generating one or more query language statements to be run against a one or more data sources, comprising the steps of:

selecting a syntax pattern corresponding to a desired function defining a default syntax pattern, and a syntax standard;

populating the syntax pattern with an argument data set associated with parameters that specify information of a desired data set from the data source and the desired function, wherein

the arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements;

identifying a functional element corresponding to the desired function and at least one syntax pattern, the functional element providing a basis for selecting the syntax pattern; and wherein, the populated syntax pattern comprises one or more query language statements having a tree query structure generated based at least in part on the parameters that specify information of the desired data set runnable against one or more data sources to return the desired data result set.

15. (Previously Presented) The computer-implemented method of claim 10, further comprising the step of identifying at least one selection variable for selecting the syntax pattern from a plurality of syntax patterns.

16. (Currently Amended) A method of generating one or more query language statements to be run against a one or more data sources, comprising the steps of:

selecting a syntax pattern corresponding to a desired function defining a default syntax pattern, and a syntax standard;

populating the syntax pattern with an argument data set associated with parameters that specify information of a desired data set from the data source and the desired function, wherein the arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the default syntax pattern to generate the one or more query language statements;

identifying the argument data set associated with the desired data set; and wherein, the populated syntax pattern comprises one or more query language statements having a tree query

structure generated based at least in part on the parameters that specify information of the desired data set runnable against one or more data sources to return the desired data result set.

17. (Original) The method of claim 16, wherein the step of identifying the argument data set includes identifying the argument data set based upon a syntax description associated with the desired function.

18. (Original) The method of claim 10, wherein the method is executed in an online analytical processing systems, a reporting system, a business intelligence system, or a data mining system.

19. (Original) The method of claim 10, wherein the step of selecting the syntax pattern includes accessing a plurality of driver modules including at least one syntax pattern, each of the plurality of driver modules corresponding to a selected database management system.

20. (Currently amended) A tangible medium having a computer readable program code embodied therein for generating one or more query language statements through an automated computer-implemented method comprising:

code for causing the processor to identify a functional element corresponding to a desired function for use in generating the one or more query language statements, wherein the functional element defines a default syntax pattern;

code for causing the processor to identify an arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements, wherein the

argument data set is mapped to positions within the default syntax pattern to generate the one or more query language statements;

code for causing the processor to select a syntax pattern corresponding to the functional element; and code for causing the processor to populate the selected syntax pattern with the identified argument data set to assemble at least one query language statement having a tree query structure generated based at least in part on the parameters that specify information of the desired data set to be run against a data source to return the desired data set.

21. (Currently Amended) A method of generating a query language statement from computer code embodied on a computer readable media comprising the steps of:

defining a syntax pattern accessible to a system for generating a query language statement having a tree query structure generated based at least in part on the parameters that specify information of the desired data set;

accessing the defined syntax pattern from the system to generate the query language statement; wherein arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements

and wherein the system does not need to be recompiled as a result of defining the syntax pattern.

22. (Original) The method of claim 21, wherein the syntax pattern is associated with a selected database management system.

23. (Currently amended) A tangible medium having a computer readable program code embodied therein for generating one or more query language statements comprising:

code for causing the processor to identify a functional element corresponding to a desired function, wherein the functional element defines a default syntax pattern;

code for causing the processor to identify an argument data set associated with parameters that specify information of a desired data set and the identified functional element, wherein the arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements;

code for causing the processor to select a syntax pattern corresponding to the functional element; code for identifying a functional element corresponding to the desired function and at least one syntax pattern, the functional element providing a basis for selecting the syntax pattern; and

code for causing the processor to populate the selected syntax pattern with the identified argument data set to assemble at least one query language statement having a tree query structure generated based at least in part on the parameters that specify information of the desired data set to be run against a data source to return the desired data set.

24. (Currently amended) A tangible medium having a computer readable program code embodied therein for generating one or more query language statements comprising:

code for causing the processor to identify a functional element corresponding to a desired function, wherein the functional element defines a default syntax pattern;

code for causing the processor to identify an argument data set associated with parameters that specify information of a desired data set and the identified functional element wherein the

arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements;

code for causing the processor to select a syntax pattern corresponding to the functional element; code for identifying the argument data set associated with the desired data set; and code for causing the processor to populate the selected syntax pattern with the identified argument data set to assemble at least one query language statement having a tree query structure generated based at least in part on the parameters that specify information of the desired data set to be run against a data source to return the desired data set.

Allowance

3. Claims 1-24 are allowable

Reason for Allowance

4. The cited arts of record, Saeki, Joji (hereinafter Saeki) US Publication No 20040039730 and Stern, Jonathan et al (hereinafter Stern) US Publication No. 20020032740 in view of Kobayashi, Susumu et al (hereinafter Kobayashi) US Patent No. 6212516, teach the query generation.

Claims 1-24 are allowable. Independent claims 1, 4-6, 10, 14, 16, 20-21 and 23-24 are allowable because the cited arts of record, Saeki, Joji (hereinafter Saeki) US Publication No 20040039730 and Stern, Jonathan et al (hereinafter Stern) US Publication No. 20020032740 in view of Kobayashi, Susumu et al (hereinafter Kobayashi) US Patent No. 6212516, do not disclose, teach, or suggest the claimed limitations of query generation (in combination with all other features in the claim),

arguments data are arranged in a vector associated with a functional component by parameter type and mapped to plurality of positions within the syntax pattern to generate the one or more query language statements;

5. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to TAREK CHBOUKI whose telephone number is (571)270-3154. The examiner can normally be reached on Mon-Fri 8:00 am to 6:00 pm EST.

7. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Neveen Abel-Jalil can be reached at 571-2724074. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

8. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/TAREK CHBOUKI/

Examiner, Art Unit 2165

03/27/2011